UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/051,274	01/22/2002.	Sunil Kunisetty	50277-2895	7586

42425 7590 11/14/2007

HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110-1089

EXAMINER
PHAM, CHRYSTINE

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
11/14/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/051,274
Filing Date: January 22, 2002
Appellant(s): KUNISETTY ET AL.

Joseph M. Olsen (Reg. No. 58,724)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed August 3, 2007 appealing from the Office action mailed October 16, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,675,354	CLAUSSEN ET AL.	6-2004
2002/0004813 A1	AGRAWAL ET AL.	1-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

- Claims 1-2, 5-6, 9, 11, 15-17, 19-21, 23-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Agrawal et al. of record (US 2002/0004813 A1, *Agrawal*) in view of Claussen et al. made of record (US 6675354 B1, "Claussen").

Claim 1

Agrawal teaches a computer-implemented method of dynamically generating web pages (see at least *request for a document, script, blocks, data source, code* paragraph [0014]; *HTML document, HTTP request* paragraph [0015]; *612*, *cached blocks 614*, *client devices 606* FIG.6 & associated text; paragraph [0061]), said method comprising:

- Analyzing a page (see at least *Web page 202* FIG.2 & associated text; *Web page 202, script, blocks* paragraph [0032]) to extract static markup text (see at least *S51-S53* FIG.5 & associated text; *portion of Web page, static, static HTML* paragraph [0031]; *pages, different users, page blocks, current weather, common zip code, partial page caching* paragraph [0060]), wherein the page includes markup text and a set of code instructions executable on a server (see at least *request for a document, script, blocks, data source, code* paragraph [0014]; *HTML document,*

HTTP request paragraph [0015]; *Web scripting, application logic, ASP, JSP* paragraph [0030]; *combination of code and static HTML, script* paragraph [0031]; *server* paragraph [0027]);

- extracting the markup text from the page (see at least paragraph [0033]; *S62* FIG.5 & associated text);
- generating a script for the page based on the set of code instructions, wherein the script does not include the markup text (see at least *scripted page, static HTML, partial page caching* paragraphs [0029]-[0036]);
- loading a copy of the markup text into shared, read-only memory (see at least *S53-S58* FIG.5 & associated text; *caching, selected number of blocks, web page* paragraph [0033]; paragraph [0034]; *static portion, cache memory* paragraph [0066]; *S59-S60, S53, S61-S62* FIG.5 & associated text; *612*, *cached blocks 614*, *client devices 606* FIG.6 & associated text; *rest of the page, shopping cart, item description, user comments, cached in memory, other users* paragraph [0061]; *shared memory cache, multiple processes, machines* paragraph [0053]; paragraph [0056]; paragraph [0064]);
- in response to each request of a plurality of requests for the page from a plurality of clients (see at least *S59-S60, S53, S61-S62* FIG.5 & associated text; *612*, *cached blocks 614*, *client devices 606* FIG.6 & associated text; *rest of the page, shopping cart, item description, user comments, cached in memory, other users* paragraph [0061]; *shared*

memory cache, multiple processes, machines paragraph [0053]; paragraph [0056]; paragraph [0064]), performing the steps of:

- instantiating a distinct instance of a JSP script on the server, wherein instantiating each instance of the script does not create another copy of the markup text (see at least *S62, S59, S60* FIG.5 & associated text);
- executing said distinct instance of the script, wherein execution of each instance of the script generates a compiled page based on the copy of the markup text that resides in shared, read-only memory, and the set of code instructions (see at least paragraphs [0014]-[0015]; *S51-S62* FIG.5 & associated text);
- sending the compiled page to a client that requested the page (see at least *S60* FIG.5 & associated text);

Agrawal does not expressly disclose said script being a servlet class. However, Claussen discloses a system and method for serving dynamically generated servlet [class] using JSP to requesting clients (see at least *210* FIG.2 & associated text; *dynamically generated HTML, JSP, Java servlet* col.1:65-col.2:21; col.5:1-45). *Agrawal* and Claussen are analogous art because they are both directed to dynamically generating scripted page (i.e., servlet class). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of Claussen into that of *Agrawal* for the inclusion of a servlet class. And the motivation for doing so would have been to allow error-handling code to be

included in the servlet class (script code) for improved debugging and maintenance of requested web pages (see at least Claussen col.14:40-col.15:26).

Claim 2

The rejection of base claim 1 is incorporated. *Agrawal* further teaches storing the markup text in a resource file associated with the application (see at least *S55-S58* FIG.5 & associated text).

Claim 5

Claim recites limitations, which have been addressed in claim 1, therefore, is rejected for the same reasons as cited in claim 1.

Claim 6

The rejection of base claim 5 is incorporated. *Agrawal* further teaches wherein the class is not loaded into the shared, read-only memory when the other instances of the application are executed (see at least *S61-S62* FIG.5 & associated text).

Claim 9

The rejection of base claim 1 is incorporated. Claussen further teaches wherein the markup text includes information to be displayed to a user and an

annotation directing a user agent how to render the information to be displayed to the user; and the markup output by the executing servlet class includes the annotation (see at least *tag handler,Java object, XSL stylesheet* col.3:30-42; *Java object, servlet, XSL stylesheet* col.5:1-col.7:35).

Claim 11

Claim recites limitations, which have been addressed in claim 9, therefore, therefore, is rejected for the same reasons as cited in claim 9.

Claims 15-17, 19-21

Claims recite limitations which have been addressed in claims 2, 5, 6, and 9 therefore, are rejected for the same reasons cited in claims 2, 5, 6, 9.

Claim 23

The rejection of base claim 1 is incorporated. Claussen further teaches wherein the servlet class includes an inner class (see at least col.12:55-col.13:10).

Claim 24

The rejection of base claim 23 is incorporated. *Agrawal* (as modified by Claussen) further teaches wherein the step of loading a copy of the markup text includes hot-loading an instance of the inner class (see at least *S53-S58*

FIG.5 & associated text; *caching, selected number of blocks, web page* paragraph [0033]; paragraph [0034]; *static portion, cache memory* paragraph [0066]; *S59-S60, S53, S61-S62* FIG.5 & associated text; *612, cached blocks 614, client devices 606* FIG.6 & associated text; *rest of the page, shopping cart, item description, user comments, cached in memory, other users* paragraph [0061]; *shared memory cache, multiple processes, machines* paragraph [0053]; paragraph [0056]; paragraph [0064]).

Claim 25

The rejection of base claim 24 is incorporated. *Agrawal* further teaches wherein the inner class comprises an array of characters (see at least *shopping cart, item description, user comments, cached in memory, other users* paragraph [0061]).

Claims 26-34

Claims recite limitations, which have been addressed in claims 23-26, therefore, are rejected for the same reasons as cited in claims 23-26.

(10) Response to Argument

Argument 1 – "Neither Agrawal nor Claussen discloses instantiating a servlet class without creating another copy of the markup text" (Brief, page 10, 2nd paragraph).

Response to Argument 1

First it should be noted that Claim 1 has been rejected under 35 USC 103(a) by the combined teaching of Agrawal and Claussen. Thus, one cannot show nonobviousness by attacking references individually (i.e., piecemeal analyzing) where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Furthermore, as has been clearly established in the Final Office Action (page 3), paragraph [0014] of Agrawal specifically reads "**dynamically generating** ... the requested **document**". As has been clearly established in the Final Office Action (page 3), paragraph [0015] of Agrawal specifically reads "The **document** may be or include ... an **HTML document**". Furthermore, as has been clearly established in the Final Office Action (page 3), paragraph [0031] of Agrawal specifically reads "in **dynamically generated pages**, a significant portion of the Web page is **static**". The same paragraph explicitly reads "**each page block is generated** by a clearly defined combination of code and **static HTML** in the script". It should be clear from these paragraphs that the requested HTML web pages (also referred to HTML documents) are generated dynamically and that they contain static HTML (i.e., **static markup text**). As has been established in the Final Office Action (page 3), paragraph [0032] of Agrawal discloses "the **partial page caching** methodologies", that is to say, Agrawal "**caches** ... a **selected number of blocks** of the document (such as Web page)" which would reduce to cost of "dynamically generating Web pages paragraph" (see paragraph

[0009]) via "full page caching ... wherein the cache memory contains ... entire Web pages" (see at least paragraphs [0010]-[0011]). Paragraph [0034] of Agrawal further discloses each cached page block (e.g., a block containing the value of a stock quote or user's local weather) is associated with an expiration time (i.e., caching properties) that is used to determine the whether the value (e.g., stock quote or local weather) contained therein is still valid, that is to say, it hasn't changed (i.e., **static**). If the value of the **cached page block** hasn't changed (i.e., **static** and/or still valid) then it can "**be retrieved and used in assembling a page** responsive to an **HTTP** request". The same paragraph further discloses "**dynamically generating those blocks that are not present** in the cache". Thus, it is clear from these paragraphs that Agrawal discloses dynamically generating HTML pages without generating (i.e., creating) another copy of the static HTML (i.e., static markup text) which as been retrieved from the cache memory. As has been clearly stated in the Final Office Action (page 4), Agrawal does not expressly disclose said dynamically generated (i.e., instantiated) document as an instance of a "servlet class" *per se*. However, as established in the Final Office Action (page 4), col.1:30-col.2:21 of Claussen discloses techniques for dynamically serving web page content (i.e., "**dynamically generated HTML**"). The same passage discloses the JSP (Java Server Page) as **web templates** that enable Java code to be embedded in **static HTML** to be served in response to a client browser request (e.g., HTTP request). The same passage further discloses translating the flat web page into the servlet, compiling the servlet to generate the servlet class, loading the generated class and invoking the servlet "to cause given (e.g., customized) web content to be

returned to the requesting browser". It is clear from this passage that the "customized web content" teaches "an instance" (that is different from other customized instances) of the generated servlet class. Moreover, col.5:1-45 of Claussen explicitly discloses translating a flat (i.e., static) web page (e.g., HTML page) using DOM to generate a web page template (e.g., HTML page) in order to generate the servlet. The same passage and FIG.2 clearly disclose loading class for the generated servlet at page translation time (when the request, i.e., HTPPRequest, for the flat web page file is accessed/hit). Step 214 of FIG.2 further discloses invoking (i.e., instantiating) the loaded servlet class. Thus, it is clear from above that Claussen discloses dynamically generating HTML pages in the context of dynamically generating an instance of (i.e., instantiating) the servlet class.

Since, Agrawal already teaches dynamically generating HTML pages without creating another copy of the static HMTL (i.e., static markup text), and Claussen teaches dynamically instantiating the servlet class from dynamically generated HTML page, the combined teaching of Agrawal and Claussen clearly teaches "instantiating a distinct instance of the servlet class without creating another copy of the markup text".

Appellants further assert, "these portions of Agrawal actually refer to retrieving cached blocks of the web page, **not to static elements within a servlet class**" (Emphasis added)(Brief, page 10, last paragraph). However, it is respectfully submitted that "static elements within a servlet class" are not recited in the claims. Appellants

further assert, "The motivation to combine does not make sense" (Brief, page 13, ongoing paragraph). However, the fact that appellants have recognized another advantage which would flow naturally from following the suggestion of the prior art cannot be the basis for patentability when the differences would otherwise be obvious. See *Ex parte Obiaya*, 227 USPQ 58, 60 (Bd. Pat. App. & Inter. 1985).

Argument 2 – "Claussen does not disclose "servlet class includes inner class" (Brief, page 13, last paragraph).

Response to Argument 2 – As has been pointed out by Appellants, the Final Office Action (page 6) cited col.12:55-col.13:10 of Claussen, which explicitly discloses inserting methodDefinition element as a child (inner element) of the root element (outer element) of the HTML document. The same passage also discloses embedding one scripting language (i.e., inner element) within another scripting language (i.e., outer element). Needless to say, two elements from two different scripting languages must be defined differently (having different classes). Thus, a class element that is embedded in another class element (e.g., servlet class) clearly teaches the "inner class".

Argument 3 – "none of the cited portions from Agrawal actually teach an inner class" (Brief, page 14, 1st paragraph).

Response to Argument 3 – As established in the Final Office Action and above, Agrawal was not relied upon to teach the "inner class". Thus Appellants' argument,

similar to Argument 1, appears to be a piecemeal analysis of the references. Appellants further assert that Agrawal does not teach "inner class that includes an array of characters". However, as established above, Claussen teaches **embedding one element** of one scripting language **into another element** of another scripting language in the same servlet. And as pointed out by Appellants, paragraph [0061] of Agrawal discloses a shopping cart (i.e., a class element) containing item descriptions and/or user comments on the items, which are cached and subsequently retrieved upon requests from multiple users. It is clear that the item descriptions and/or user comments are text strings (i.e., "array of characters"). Thus, it is obvious that the shopping cart class (defined in one scripting language) containing the cached static item descriptions (which are the same for all users) can be embedded in the servlet class (defined in another scripting language), and the shopping cart class can be instantiated for each different user requesting an instance of the shopping cart.

Argument 4 – Agrawal and Claussen do not teach "accessing the set of markup text in a shared, read-only memory when the code from the first instance of the application is executed".

Response to Argument 4 – As established above in Response to Argument 1, paragraph [0034] of Agrawal explicitly discloses that if the value of the **cached page block** hasn't changed (i.e., **static** and/or still valid) then it can "**be retrieved and used in assembling a page** responsive to an **HTTP** request". The same paragraph further discloses "**dynamically generating those blocks that are not present** in the cache".

It is clear that Agrawal's cache memory teaches the "shared, read-only memory". Furthermore, as discussed above, FIG.2 of Claussen clearly disclose loading class for the generated servlet at page translation time (when the request, i.e., HTPPRequest, for the flat web page file is accessed/hit). Step 214 of FIG.2 further discloses invoking (i.e., instantiating) the loaded servlet class. Thus, it is clear the Agrawal's static markup text (i.e., HTML) is retrieved (i.e., accessed) from the cache memory (i.e., shared, read-only memory) to be assembled with the dynamically generated HTML when the Claussen's servlet is invoked.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Chrystine Pham

Conferees:

SPE Tuan Dam, AU 2192

TUAN DAM
SUPERVISORY PATENT EXAMINER

SPE Eddie Lee, Tech Center 2100

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER